Image-based recognition of natural disasters and other events that might require human assistance

A DEGREE THESIS SUBMITTED TO THE FACULTY OF ESCOLA TÈCNICA D'ENGINYERIA DE TELECOMUNICACIÓ DE BARCELONA
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Núria Marzo i Grimalt

In partial fulfillment of the requirements for the degree in Telecommunications Technologies and Services Engineering and the degree in Engineering Physics

Advisors: Antonio Torralba and Xavier Giró

Cambridge, MA

Spring 2019

Abstract

When some type of unexpected event occurs anywhere in the world such as an earthquake or a flood, many images are uploaded to social media platforms like Twitter. Extracting information from these images can be useful to quantify the damage or the magnitude of the disaster. Deep learning models applied to image recognition have proven to be capable of performing many tasks like classifying objects, faces or places. However, they fail when dealing with images containing objects or structures that have been damaged. This issue could be solved by training deep learning models to recognize these kind of scenarios. Yet, it would require a large labeled dataset of images containing examples of damaged places.

In this work we present *Incidents*, a database of images showing natural disasters or other types of events that might require human assistance. It is a diverse dataset where all incidents appear in many different places. We also introduce a deep learning model that, given an image with some catastrophic event, classifies the incident and the scenario. Finally, we propose some interesting applications for these types of image recognition models.

Keywords: deep learning, machine learning, image recognition, computer vision, scene recognition, humanitarian, natural disasters.

Resum

Quan succeeix algun tipus d'esdeveniment inesperat, com un terratrèmol o una inundació, es comparteixen moltes imatges a xarxes socials com Twitter. Extreure informació d'aquestes imatges pot ser útil a l'hora de quantificar el dany o la magnitud del desastre. Els models d'aprenentatge profund aplicats al reconeixement d'imatges han demostrat ser capaços de realitzar moltes tasques com classificar objectes, cares o llocs. No obstant, fallen quan es tracta d'imatges que contenen objectes o estructures que han estat danyades. Aquest problema es podria resoldre mitjançant l'entrenament de models d'aprenentatge profund per reconèixer aquests tipus d'escenaris. Tanmateix, es necessitaria una base de dades àmplia amb imatges etiquetades que continguin exemples de llocs danyats.

En aquest treball presentem *Indicents*, una base de dades d'imatges on es mostren desastres naturals o altres tipus d'esdeveniments que poden requerir assistència humanitària. També introduïm un model d'aprenentatge profund que, donada una imatge amb algun esdeveniment catastròfic, classifica l'incident i l'escenari. Finalment, proposem algunes aplicacions interessants per a aquest tipus de models de reconeixement d'imatge.

Paraules clau: aprenentatge profund, intel·ligència artificial, reconeixement d'imatge, visió per computador, reconeixement d'escena, humanitària, desastres naturals.

Resúmen

Cuando ocurre algún tipo de evento inesperado en cualquier parte del mundo, como un terremoto o una inundación, se comparten muchas imágenes en redes sociales como Twitter. Extraer información de estas imágenes puede ser útil para cuantificar el daño o la magnitud del desastre. Los modelos de aprendizaje profundo aplicados al reconocimiento de imágenes han demostrado ser capaces de realizar muchas tareas como clasificar objetos, caras o lugares. Sin embargo, fallan cuando se trata de imágenes que contienen objetos o estructuras que han sido dañadas. Este problema podría resolverse mediante el entrenamiento de modelos de aprendizaje profundo para reconocer estos escenarios dañados. Aún así, se requeriría una amplia base de datos con imágenes etiquetadas que contengan ejemplos de lugares dañados.

En este trabajo presentamos *Indicents*, una base de datos de imágenes que muestran desastres naturales u otros tipos de eventos que puedan requerir asistencia humanitaria. También introducimos un modelo de aprendizaje profundo que, dada una imagen con algún evento catastrófico, clasifica el incidente y el escenario. Finalmente, proponemos algunas aplicaciones interesantes para este tipo de modelos de reconocimiento de imagen.

Palabras clave: aprendizaje profundo, inteligencia artificial, reconocimiento de imagen, vision por computador, reconocimiento de escena, humanitaria, desastres naturales.

Acknowledgments

First of all, I would like to express my deepest gratitude to my advisor Prof. Antonio Torralba for his support, motivation and immense knowledge. He granted me an opportunity to work in his research laboratory beside such a passionate community that is working on the latest challenges in computer vision.

Secondly, I would also like to thank Prof. Àgata Lapedriza for her help and advice. Her guidance helped me during all the research and writing of this thesis and has been a great support to carry out this project.

Additionally, I would like to acknowledge Dr. Ferda Ofli and Dr. Muhammad Imran of QCRI (*Qatar Computing Research Institute*) and Aritro Biswas of *Oracle* and *MIT Alumni* for their work in this project.

I also want to thank the *Massachusetts Institute of Technology*, the *Centre de Formació Interdisciplinària Superior* and fundació *CELLEX* for their economical support on this project.

Last but not least, I couldn't be any more thankful for my friends and family's continuous encouragement. Specially Jose, Martí and Oriol, whose company and care has been priceless throughout this experience in Boston. Moltes gràcies per tot.

Contents

List of Tables

List of Figures

1 Introduction

In this section the motivation behind the idea of the project is going to be explained, as well as the first experiments performed to end up defining the problem we wanted to solve.

1.1 Motivation

In 2017 two major hurricanes hit the United Stated of America: hurricane Harvey and hurricane Irma. The first one hit the state of Texas on the 25$^{\text{th}}$ of August and its effects (e.g. flooding and rain) lasted until the end of the month. Hurricane Irma hit the state of Florida on September 9$^{\text{th}}$ and its effects lasted for a week until September 15$^{\text{th}}$.

According to the work of Jahanian et al. (2018) [11] there was a significant use of Twitter during these events. For Harvey almost 2 million tweets were crawled within a 500-mile radius of Houston and 25% of them were tweeted during the first hour and a half after the hurricane had hit the city. For Irma, 14.4 million of tweets were collected within a 500-miles radius of the center of Florida and 3.4 million of them were uploaded during a period of one hour and a half on September 12$^{\text{th}}$. In their article they conclude that when traditional 911 service and cellular communication infrastructure had 80% of cell sites down, thousands of tweets with geo-tagging information were sent from these areas carrying a lot of information about the catastrophic event; information that could have been used by the police authorities to respond quickly and efficiently to the incident.

Many situations like these take place around the globe and this issue becomes important in countries where the police authorities and the emergency services do not usually cooperate very well or the cellular communication infrastructure is in a poor state. Being in a world where information is highly valuable, we realized that there was a lot of data uploaded every day on social media that could be used as a source for a lot of humanitarian work.

The main idea that came to our mind was to have a model that could recognize incidents from images uploaded to platforms like Twitter. And, because many tweets have geo coordinates, we could represent any incident happening around the world instantaneously. Another important feature we wanted to extract from the images was the place where the disasters happened. To perform this last task there are some scene-centric classifiers but we had to test if they would work with damaged places, objects and structures.

1.2 First experiments

We took one of the state-of-the-art scene-centric classifiers: the Places365 Classifier developed by Zhou et al.[18] and checked if it could identify different places with some natural disaster or incident on them. The results are shown in Table 1.1.

In the first image the most important characteristic is that the place (in this case a field) is on fire but the label that is assigned to the image is *volcano* with a high confidence. In the second image we can see a highway flooded with many stopped cars on the side of the street, the classifier in this case tells us that it is a *parking lot* as it identifies a lot of cars. Lastly, in the third image a house covered with snow is shown and the classifier labels this picture as an *igloo*.

Image	**Place Type and Label**
	• Type of environment: OUTDOOR. • Place: VOLCANO (0.717)
	• Type of environment: OUTDOOR. • Place: PARKING LOT (0.568)
	• Type of environment: OUTDOOR. • Place: IGLOO (0.462)

Table 1.1: Images with some incidents and the place type and label according to Places365.

After seeing these results there was no doubt that new models needed to be trained to perform

this kind of tasks as the state-of-the-art ones do not work in all cases, especially when they are dealing with damaged places.

As explained in the previous section (1.1), the first idea suggested was to have a classifier that given an image could identify the incident happening and the scenario where it was taking place.

To have this type of classifiers, we needed an image database that could be used to train these models. That is why we propose the *Incidents* database. A database with images classified with two different tags: an incident tag and a place tag. As shown in the matrix of Figure 1.1 we expect to have an incident, for example a flood, taking place in many different scenarios such as a forest road, a field, a house, a downtown or the coast; as well as many other incidents like burning, fog, snow covered or car accident.

As a final product we would like to have a database that resembles the matrix in Figure 1.1 but with many more incidents as well as places, and to have as many images as possible in every mixed category of Incident-Place. With this type of databases we could train a classifier that, given an image, could predict its disaster and its scenario.

Figure 1.1: Incident-Place Matrix.

2 Theoretical Background

In this chapter some of the main theoretical concepts used in this project are will be explained. First, it will focus on the image classification problem. Next, convolutional neural networks will be introduced and lastly, some features of residual networks will be exposed.

2.1 The Image Classification Problem

The image classification problem is one of the core issues in the field of computer vision. It consists in assigning an input image one label from a fixed set of categories. This type of task may be relatively trivial for a human to perform, but when performed by a machine it can face many challenges. Some examples of these main challenges that a computer vision algorithm might face are:

- Any object can be oriented in many different ways with respect to the camera.
- The illumination conditions might change with the consequent drastic modification on the value of the pixels.
- The object to recognize can be occluded and only a small portion of it can be visible.

Therefore, a good model for image classification has to handle all these challenges and more shown in Figure 2.1. [2]

Many different methods can be tried to solve this problem but the data-driven approach is the one that has proven more successful and scalable. This approach can be formalized as follows:

- There is an input that consists of a set of N images, each labeled with one of K different classes. These images are the *training set*.
- The training set has to be used for a classifier to learn how every class looks like.
- The quality of the classifier has to be evaluated using a set of images that the classifier has never seen before. Then the label given to the new images has to be compared with its true label, also known as *ground truth*.

Figure 2.1: Example of challenges that the classifiers can face.

2.2 CNNs: Convolutional Neural Networks

One of the main algorithms used nowadays to solve the image recognition problem are convolutional neural networks. These artificial neural networks were first introduced by Fukushima in 1988 [7] but were not very used due to the limits of computation hardware. In 1990, Yan LeCunn et al. [14] applied a gradient-based learning algorithm to CNNs and obtained successful results for the handwritten digit classification problem.

Over the years, researchers improved these algorithms and the computation power increased. In 2012, Krizhevesky et al. presented AlexNet [12], a seven layer CNN that won the most difficult ImageNet [4] challenge for visual object recognition called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). This was a significant breakthrough in the field of deep learning and computer vision. [5]

These architectures are a type of neural networks commonly used in computer vision for classification and segmentation problems. They are very similar to ordinary neural networks but in CNNs there is the assumption that the inputs are images.

Ordinary neural networks are made up of neurons that have learnable weights and biases. They receive an input (vector) and transform it through a series of hidden layers. Every layer is made up of a set of neurons that is fully connected to all neurons of the previous layer. Each neuron of the layer operates independently and does not share connections. The last layer is called the *output layer*. In Figure 2.2 there is a visualization of how neural networks and their connections look.

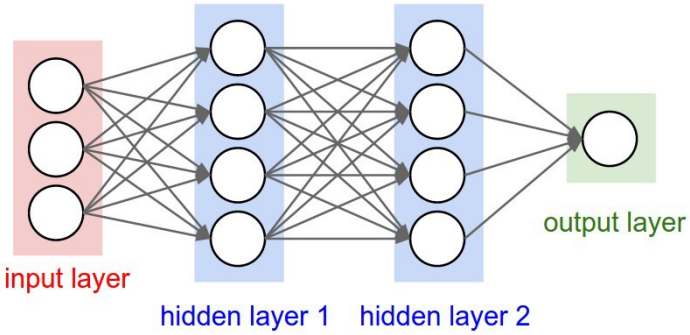


Figure 2.2: Example of a typical neural network structure.

On the other hand, convolutional neural networks consist in a set of layers transforming some input given into an output and the name "convolutional" indicates that the network employs a mathematical operation called convolution in at least one of its layers. The architecture of a typical CNN (Figure 2.3) is structured as a series of stages. The first stages contain of two types of layers: convolutional layers and pooling layers, and the last stages contain fully connected layers. [13]

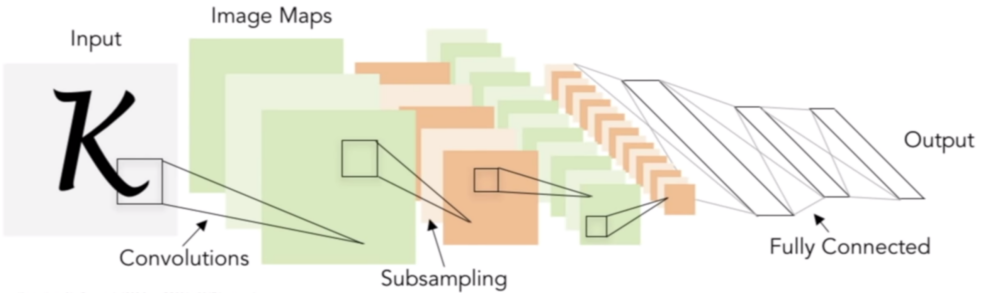


Figure 2.3: Example of a typical convolutional neural network structure.

A brief explanation of these layers will be given as we will reference them in the following sections:

- **Convolutional Layers**: These layers consist in a set of learnable filters. Each of the filters is smaller than the input image spatially (height and width) but extends through the full depth of the input volume (e.g. a typical filter has dimensions of $5x5x3$ where the depth is 3 because the input images have the three color channels RGB). During the forward pass, each filter slides across the width and height of the input volume and computes dot products between the values of the filter and the input at any position. A 2-dimensional activation map is computed as the filter slides over the width and height of the input volume. It gives the responses of the filter at every spatial position. The network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or some color in the first layers, or more complex patterns in higher layers of the network. Figure 2.4 shows an example of how these filters look.

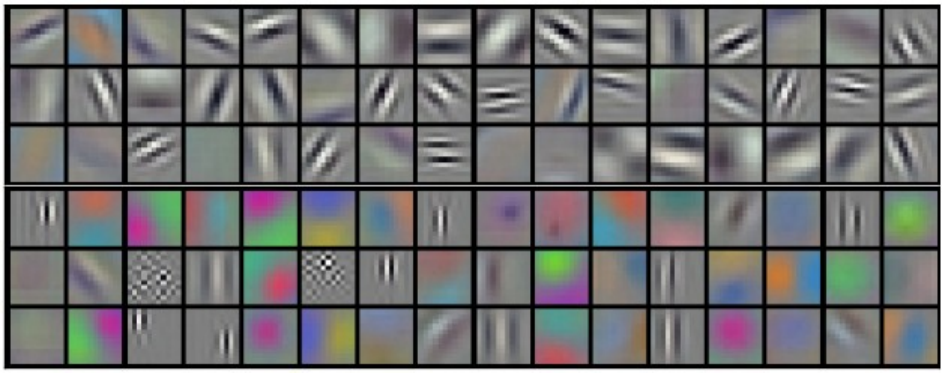


Figure 2.4: Example of filters learned by AlexNet. The filters have a size of $11x11x3$. [12]

  A non-linearity is generally applied at the output of convolutional layers, normally being the function $f(x) = max(0, x)$. This step is known as the *activation layer*. Specifically, if the function $f(x)$ is used the layer is called a *ReLU* (Rectified Linear Units).

- **Pooling Layers**: This layer divides the input in patches and takes the maximum value

for each patch. It only performs the operation in the spatial dimension. The number of parameters to learn is reduced as a result of down sampling the input data.

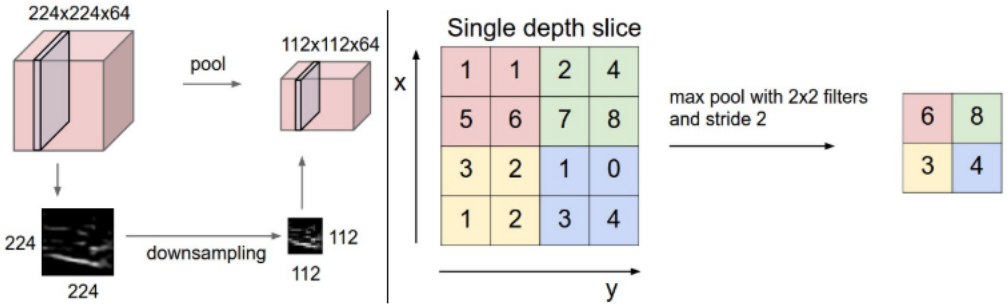


Figure 2.5: In the left there is an example of a pooling layer for an input of size $224x224x64$. In the right there is the most common pooling operation: the max.

- **Fully Connected Layers**: In this kind of layers, neurons in adjacent layers have full pairwise linear connections, though neurons within a layer are not connected.

Moreover, an interesting parameter to understand when training neural networks is the loss function or cost function. This function quantifies the agreement between the predicted scores and the ground truth labels. We can understand the training of a classifier as an optimization problem in which the weights of the different layers are updated in order to minimize the loss function.

2.3 Residual Neural Networks

As CNNs got deeper and computation power increased, accuracy was expected to reach higher values. However, a degradation problem was observed: as the network depth increased, the accuracy got saturated and then degraded rapidly. He at al. from *Microsoft Research* came up with a solution introducing the deep residual learning framework. [9]

When, in a network, there are several layers stacked with an input x and an output y, $x \Rightarrow y$ can be directly mapped with a function $H(x)$. In the residual networks, instead of hoping that the layers fit a desired underlying mapping, we can explicitly let these layers fit a residual mapping. This mapping is formally denoted as $F(x) := H(x) - x$ and the original mapping recasts into $F(x) + x$.

He et al. [9] hypothesize that it was easier to optimize the residual mapping rather than optimizing the original unreferenced mapping. The formulation of $F(x) + x$ can be realized by feed forward neural networks with shortcut connections. These blocks are shown in Figure 2.6.

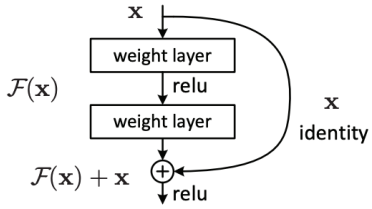


Figure 2.6: Residual Unit.

Shortcut connections are those skipping one or more layers. In this case the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers. These shortcut connections do not add extra parameters or computational complexity. The entire network can still be trained by Stochastic Gradient Descent with back-propagation. An example of a residual network compared to a plain network can be seen in Figure 2.7 where there are many residual blocks.

Deep residual networks will be used for our classification task.

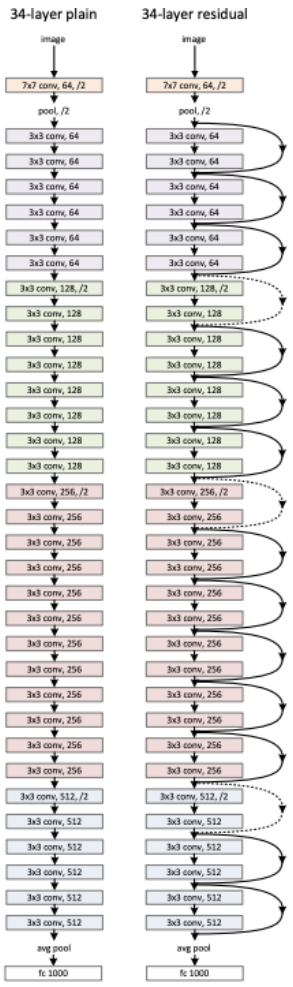


Figure 2.7: A 34-layer plain network versus a 34-layer residual network.

3 Related work

Many tech companies try to use its products to do some kind of humanitarian action. The approach of dealing with different incidents is not new and many companies have some related work.

One example is *Microsoft Research* with their project *AI for Good* [1]. One of the aims of Microsoft is to try to reduce the resistance and fear that some have to A.I. using it to tackle some humanitarian issues such as: disaster response, needs of children, refugees and displaced people and human rights. Focusing on the disaster response application, it is aimed at predicting when and where a disaster may strike and with computer vision technologies it analyzes images of places destroyed by some catastrophic event and helps agencies to improve their response in terms of speed and safety.

Another company that has taken some steps in disaster detection and response is *Facebook Artificial Intelligence* with the Facebook Disaster Maps [17]. These maps are able to provide information about how the population moves and where they check safe during some kind of catastrophe. They gather information in real time from Facebook users that share their location and create density maps, movement maps and safety check maps. However, this is limited to the segment of the society that uses Facebook and that is sharing its localization when this type of events occur.

Facebook Artificial Intelligence has also another project: From Satellite Imagery to Disaster Insights [6]. In this initiative they try to analyze satellite images and identify the areas with maximum damage after some kind of disaster. It compares data captured both before and after a disaster and with a deep model trained on general roads and buildings datasets it can identify areas of maximum damage. The researchers also propose a new metric to help quantify the detected changes: Disaster Impact Index (DII), normalized for different types of features and disasters.

Finally, in the field of scene recognition one of the main important projects to highlight is the Places365 Database [18]. It consists of a database of more than 10 million images distributed in 365 different scene categories. The main difference between Places365 and the research we are conducting is that we intend to recognize places with a damaged environment, a task in which the classifier trained with Places365 fails.

4 Incidents Database

In this chapter the building of the Incidents dataset will be presented. The first section will discuss the creation and transformation of the image categories of the dataset. The second section will discuss the construction of the dataset with its main steps. And the third and final section will show some of the main statistics of the database.

4.1 Categorical Space

The categorical space is divided into two main independent kinds: the incidents and the places. Both of the them have suffered transformations while developing the database due to different reasons. The aim with this section is to explain the transformation that the categories have suffered and the reasons.

The first definition of the categorical space had 233 incidents grouped by type. The different types used were: damaged, natural disaster (caused by nature), transportation and nuclear accident, man-made incident (incidents caused directly by humans) and people attributes (incidents that involved people)[1]. The amount of categories per type is described in the Table 4.1.

All these different incidents were very fine grained and most of them overlapped with others. For example, in transportation accidents we had *van disaster*, *van accident*, *van crash*, *van wreck* and *van incident*, those could be compressed into a less fine grained category called *van accident*. Additionally, all the images had to be annotated with crowd-sourcing platforms and any category involving directly humans in uncomfortable situations such as *bleeding people*, *terrorist attack* or *killing* could be a problem when working with services as Mechanical Turk of Amazon. Therefore, we decided to remove all of the human incidents.

After all these changes, the categorical space had 43 final incident categories which can be seen in Table 4.2. Some example images and definitions for the incident categories can be found in Appendix B.1.

	Damaged	**Natural Disaster**	**Man-Made Incident**	**Transp. Accident**	**People Attributes**	**TOTAL**
Old Categories	12	49	28	118	26	**233**
New Categories	9	22	-	12	-	**43**

Table 4.1: Number of incident categories by type.

[1]The name of the old categories grouped by type is specified in Appendix A.1

Type	Categories
Damaged	damaged, flooded, dirty-contaminated, blocked, collapsed, under construction, on fire, with smoke and burned
Natural Disaster	ice storm, drought, dust-sand storm, dust devil, thunderstorm, tropical cyclone, tornado, fire whirl, derecho, heavy rainfall, hailstorm, earthquake, landslide, mudslide mud-flow, volcanic eruption, snow-slide avalanche, sinkhole, storm surge, wildfire, fog, rock-slide rock-flow and snow covered
Transportation and Nuclear Accidents	airplane accident, car accident, train accident, bus accident, bicycle accident, motorcycle accident, van accident, ship-boat accident, truck accident, oil spill, nuclear explosion and traffic jam

Table 4.2: Incident categories grouped by type.

For the categories concerning places we started from the Places365 database [18], a 10 million images database with 365 scene categories widely used in the field of scene recognition. After looking at them in depth we immediately removed the ones that involved an indoor scene and ended up with 101 different categories inherited from Places365 and 17 added by the incidents team. The next step was to see if any of them involved any kind of humans in an uncomfortable situation such as *immigration rids* or *refugee camps* or any type of war or military scenery. Also, most of the classes were very fine grained and it didn't make sense for us to try to find incidents in very specific places because the database would be small and with very few applications. The final result was the 49 place categories listed in Table 4.3. Some example images and definitions for the place categories can be found in Appendix B.2.

Type	Categories
Place	badlands, beach, bridge, building facade, building outdoor, cabin outdoor, coast, construction site, dam, desert road, desert, downtown, excavation, farm, field, fire station, forest, forest road, gas station, glacier, highway, house, industrial area, junkyard, lake natural, landfill, lighthouse, mountain, nuclear power plant, ocean, oil rig, park, parking lot, pier, port, power line, railroad track, religious building, residential neighbour, river, sky, skyscraper, slum, snowfield, sports field, street, valley, village and volcano

Table 4.3: Place categories.

4.2 Constructions of the database

The construction of the Incidents Database is composed of three different steps. It starts with querying and downloading images, then removing any duplicates within each category and finally labeling these images with ground truth categories. The detail of each step is introduced in the following sections.

4.2.1 Step 1: Downloading images using queries of mixed keywords

The aim with the downloading process was to have images for all of the incidents in different places. With our 43 incidents we wanted to have images of a specific incident in the 49 different places. If we combine every incident category with every place category we have $2107 = 43 \cdot 49$ of mixed keywords. Using those mixed keywords, queries were arranged with synonyms of the categories and connectors between the incident and place keyword. For example the images of the mixed keyword of *(flooded, street)* were downloaded from queries such as: *flooded street*, *flood in street*, *river flood in street*, *coastal flood in street*, *flooded alley*, *flood in alley*, *river flood in alley*, *coastal flood in alley*. For these 2107 mixed keywords of *(incident, place)* we computed 10178 different queries of images to be downloaded.

Images for every query were downloaded using a common online search engine: Google Images. The number of images downloaded before any processing was 2 427 161, and we tried to download 400 images for every query even though in some cases it was not possible.

Note that we computed queries for every mixed category, which means that some of the queries represented some incidents in an unlikely place (e.g. boat accident on highway or snowstorm in the ocean). The decision to create queries for every mixed category even though some of them could be implausible was made because in early steps of the dataset creation, the important task was to gather as many images as possible. Some of the images downloaded with these queries can still be valuable incident images as shown in Figure 4.1 where, although the query used is *snow storm in ocean* we can still download images that represent a snow storm in unconventional places like a beach or the coast.

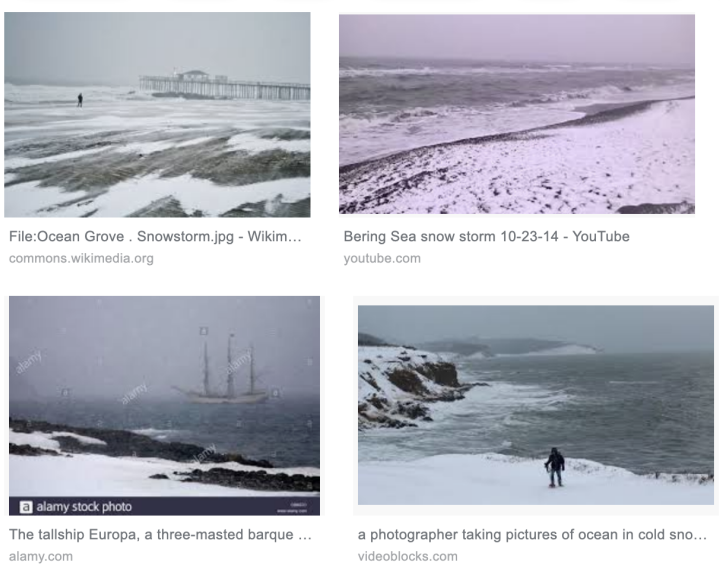


Figure 4.1: Set of four images that appear if you search *snow storm in ocean* with Google Images.

4.2.2 Step 2: Duplicate Removal

The next step was to remove any duplicates. Assuming the fact that an image could have more than one incident (e.g. *on fire* and *with smoke*), we decided we could have duplicates between different categories. Therefore, we only removed the duplicates between the synonym queries (e.g *flooded street*, *flood in street*, *river flood in street*, *coastal flood in street*, *flooded alley*, *flood in alley*, *river flood in alley*, *coastal flood in alley*). In order to achieve it, we followed some steps:

1. Using a classifier trained with images from Places365 [18] we computed the features of each image taking the values of the output layer of the neural network.
2. The images were classified into different groups according to their mixed category *(incident, place)*.
3. In each of the groups, using a radius-based nearest neighbor method, we found the nearest neighbors for every image according to the features and the euclidean distance between the neighbors.
4. After performing some visualizations shown in Figure 4.2 a threshold was decided and any images that had a distance in score lower than our threshold were considered duplicates.

The main thresholds considered were 1, 1.5 or 2. We performed the duplicate removal task for each of the thresholds and the results of the number of duplicated found are in Table 4.4.

Threshold	**Images removed**	**Images Left**	**% of removals**
1	549681	1877480	22.65%
1.5	563850	1863311	23.23%
2	595329	1831832	24.53%

Table 4.4: Duplicate removal for different thresholds.

The final threshold used was 1.5. To set this threshold we visualized some images with their nearest neighbors and the distance in the score between them as shown in Figure 4.2. Many figures like this were represented and discussed among the team and, as represented in the second image, many of the duplicates had a distance higher than 1 but lower than 1.5.

With this threshold, 23.23% of the images were considered duplicates and we were left with 1 863 311 unique images. The next step was the annotation task.

Figure 4.2: Nearest neighbors of two images with its distance. In the top row there is the image and in the bottom row there are its nearest neighbors and the respective distance.

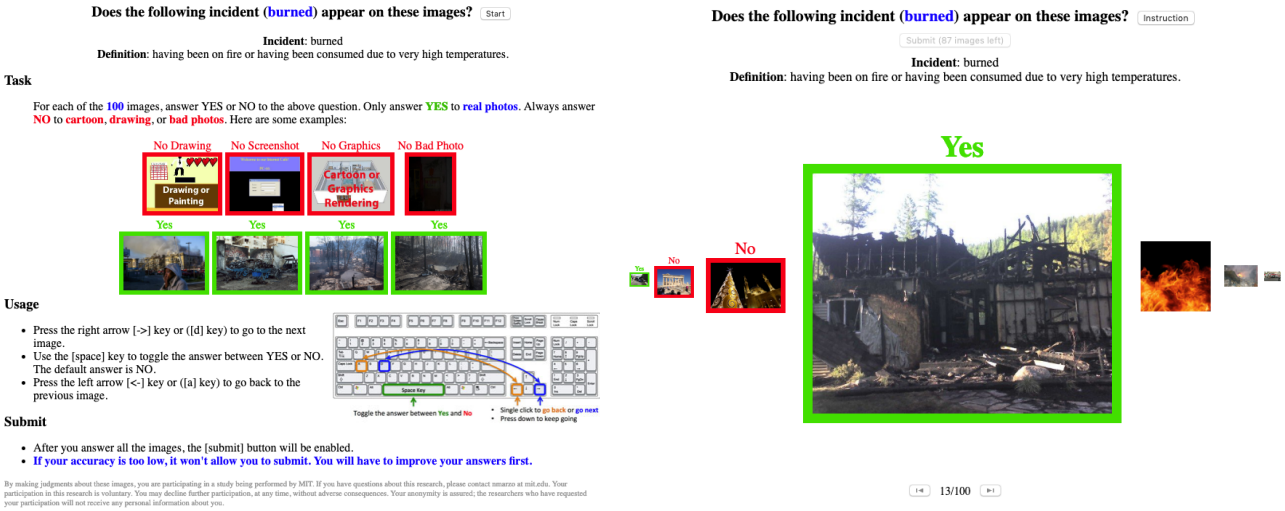
4.2.3 Step 3: Labelling images with ground truth category

The verification of the ground truth label of the images was done by crowd-sourcing the task to Amazon Mechanical Turk (AMT)[2]. We divided the labelling task into two parts: the incidents annotation and the place annotation. The first task performed was the incidents annotation.

In Figure 4.3 it appears the experimental paradigm used. First, AMT workers were given instructions relating to a particular incident category at a time (e.g. burned), with a definition and 4 sample images belonging to the category. As an example, Figure 4.3a shows the instructions for the category burned. Then the workers performed a verification task for the corresponding category. Figure 4.3b shows the AMT interface for the verification task. The experimental interface displays a central image, on the left there is a smaller version of the

[2]https://www.mturk.com/

images the worker has already responded and on the right the images the worker will respond to next. The default answer of the central image is NO.



(a) Instructions page.

(b) Verification page.

Figure 4.3: Interface used for the annotation.

The annotation task was divided in HITs of 100 images: 85 images of the ones we downloaded, 10 positive control samples and 5 negative control samples. The accuracy of the workers was checked using these 15 control samples and had to be greater than 80%. If the accuracy was lower, a message of alert appeared ("Your accuracy is too low! You are not allowed to submit. Click [Cancel] to refine the results.") and the workers were unable to submit their HITs.

The results from this task can be seen in Table 4.5.

Images annotated	**Positive images**	**% of positives**	**Accuracy of Workers**
798316	193648	26.08%	94.28%

Table 4.5: Results of the Incidents Annotation with AMT.

The main goal was to have more than 3000 labeled images per incident category. That is why, even though we had approximately 1 860 000 images in total, we only annotated 800 000 until we reached our goal. Of these 800 000 images only 193 648 were annotated as positive examples of a disaster as it is shown in Table 4.5. The distribution of images per incident category is the one shown in Figure 4.4. Note that not all the categories have more than 3000 images. The reason is that, although we annotated all of the images we had for these categories, the percentage of positive was low.

The annotation of the places labels was slightly trickier because we discussed that it could be done in different ways:

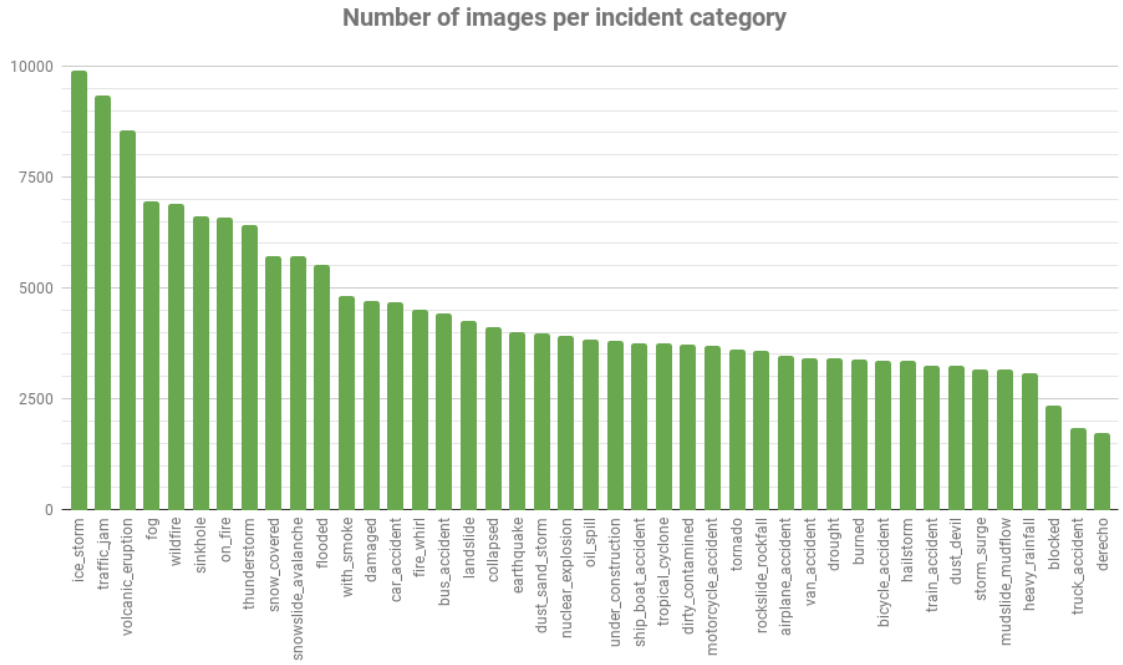


Figure 4.4: Distribution of the number of images per incident category after the annotation.

1. Annotating every image according to the place query they had been downloaded with.
2. Forward them to the Places365 classifier [18] and annotate them according to the category they were labeled with the classifier.

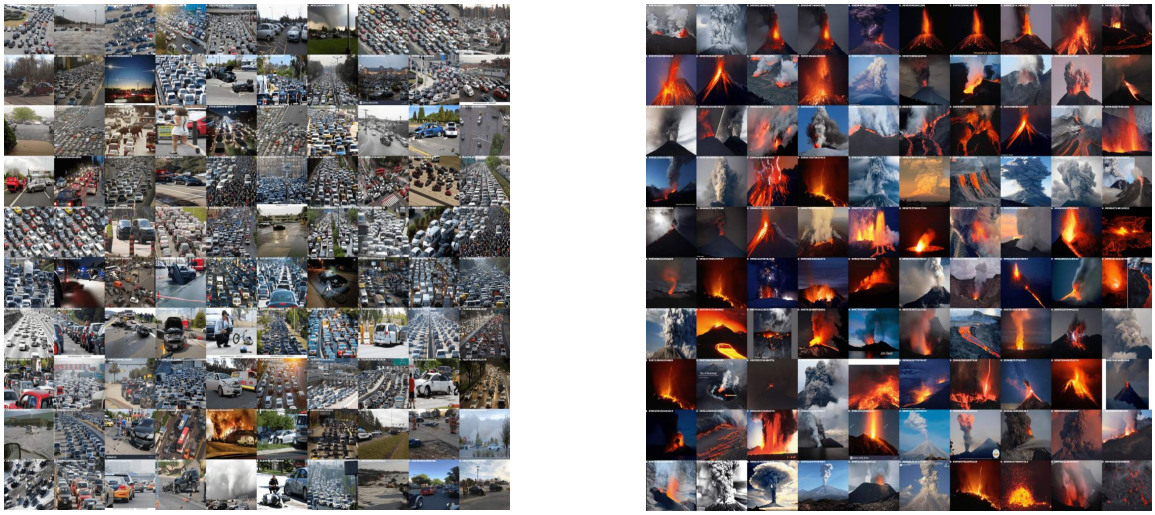
To analyze the first approach we visualized a random sample of 100 images for every place category. An example of this grids is shown in Figure 4.5. In these two example grids we can see that a lot of images in the forest grid are actually forests (Fig. 4.5a), but with dam the number of correct images is lower (Fig. 4.5b). We analyzed all the grids for our place categories and many of them had many correct images so annotating using the place category of the query seemed a reasonable idea.

Then, to analyze the second approach we used the Resnet18 Places365 Classifier with our images, grouped them by the label and sorted them by score. The results of the labels *parking lot* and *volcano* are shown in Figure 4.6. The classifier does a very good job with the volcano images (Fig. 4.6b) whereas the parking lot images appear to be streets or highways 4.6a). So as well as with the first approach some of the categories had a lot of correct images and some had not. Therefore, we could conclude that the Places365 classifier did a good job on some categories.

Finally, we decided to mix the two approaches and proceed with the annotation in the following way:

(a) Forest. (b) Dam.

Figure 4.5: Random sample of 100 images that were downloaded using the query forest or dam.



(a) Parking Lot. (b) Volcano.

Figure 4.6: Images labeled as street or volcano by the Resnet18 Places365 Classifier sorted by score.

1. All the images labeled in the disaster annotation as *volcanic eruption* were labeled directly with the place category *volcano*.

2. The images were represented in a matrix where the rows corresponded to the places in the queries and the columns to the places classifier labels. All the images that had the same label according to the query and the places classifier (images in the diagonal) were annotated directly with this place category without AMT workers.[3]

[3]To see the matrix visualization of the query labels versus the place classifier labels see Figure C.1 at Appendix C

3. Using Amazon Mechanical Turk, workers annotated all the other images using the first approach presented. They had to check if the images represented the place that was in the query they were downloaded with.

The annotation of the places followed the same interface as the incidents shown in Figure 4.3 and 15 control images were used. The workers only could sent the HIT if their accuracy on these control images was higher than 80%. The main results are shown in Table 4.6. With all the images annotated only a 34.09% were considered positives and the accuracy of the workers was 95.05%.

Images annotated	**Positive images**	**% of positives**	**Accuracy of Workers**
193648	66013	34.09%	95.05%

Table 4.6: Results of the Places Annotation with AMT.

Finally in Figure 4.7 there is a plot of the distribution of the places with the images labeled as positive in the annotation process. The categories with more images are *volcano*, *building outdoor*, *highway* and *forest* with more than 3000 images each. Those are scenarios very common for a lot of incidents. In the other hand, categories with less images are *valley*, *village* and *park*.

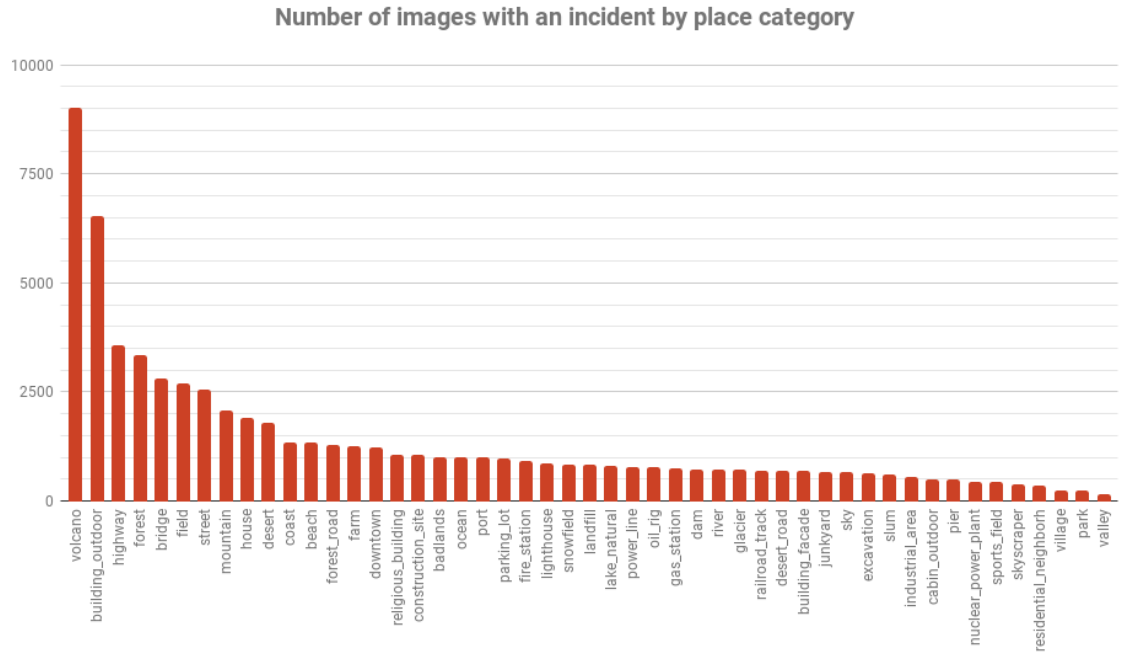


Figure 4.7: Distribution of the number of images per place category after the annotation.

4.3 Statistics of the final dataset

The final dataset has 193 648 images of two different kinds:

1. 66 013 images with both an incident label and a place label.
2. 127 635 images with only an incident label.

The distribution of images according to the incident label can be seen in Figure 4.8 and the distribution according to the places labels is in Figure 4.9.

The dataset has 43 incident classes and 49 places classes. In addition, two other categories were added to the set of incidents and places. These categories added are *no place* and *no disaster*. It is important to have a *no disaster* category in order to differentiate between images with unexpected events and normal day to day images. We didn't want a classifier that every time that it saw a car it detected a *car accident* so it was important for us to have images that contained no incidents. The *no place* category was necessary in order to differentiate between images that represented a place, for example a city or images that represented an object like a flower or a food plate.

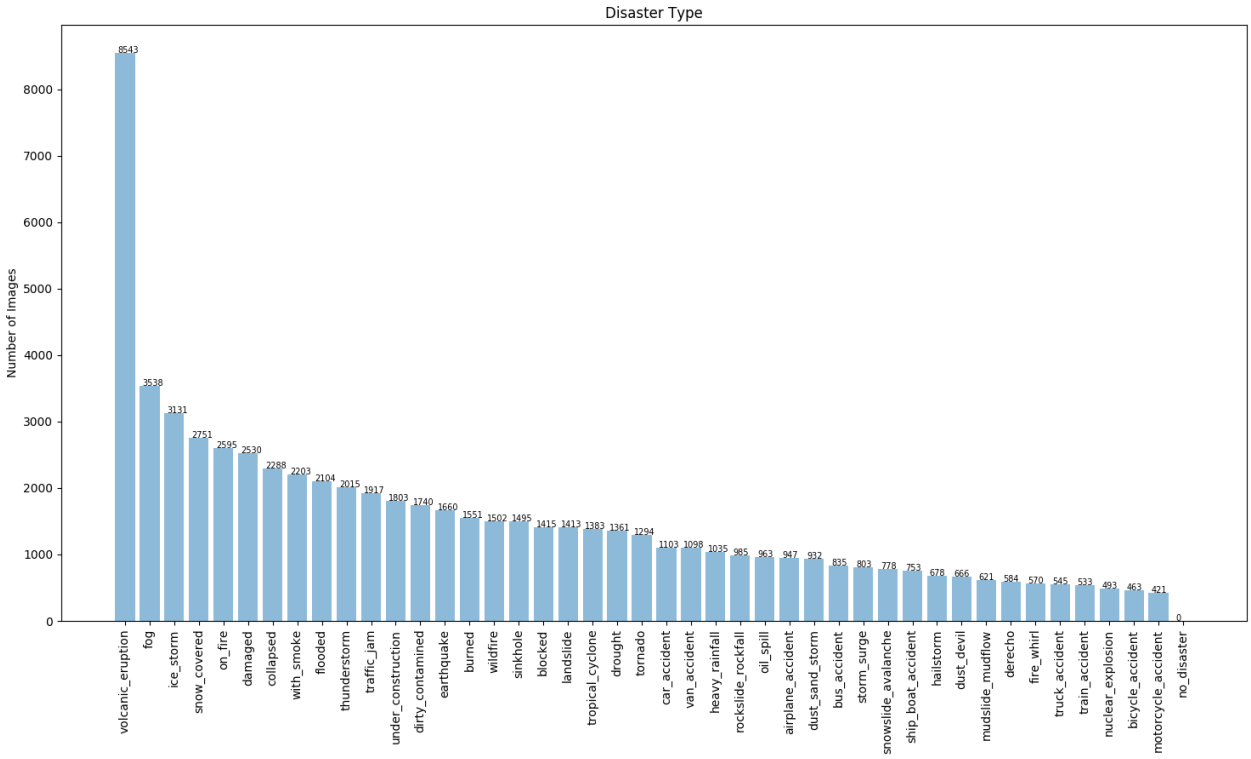


Figure 4.8: Distribution of the images in the 66 013 images part of the dataset by its incident category.

All of our 193 648 images contain different types of incidents, therefore we needed new images that contained no disasters. We decided to use images from the Places365 Database [18]. The

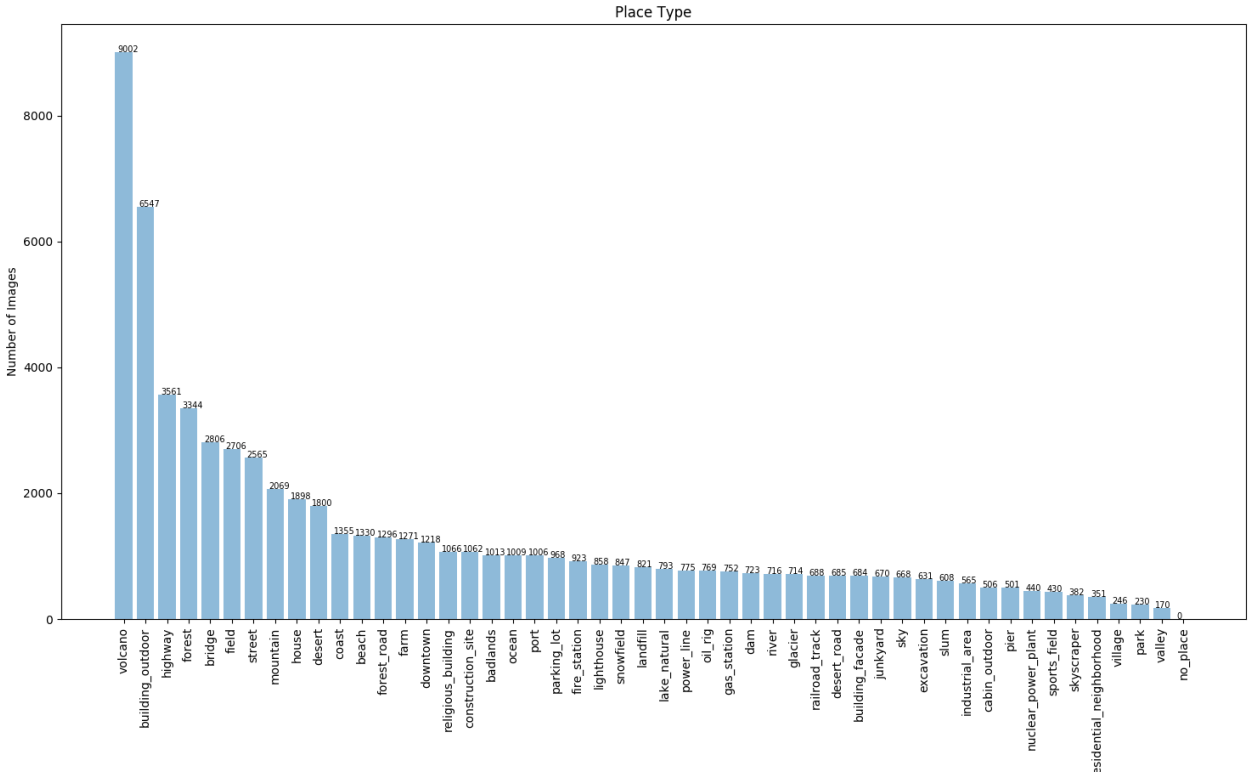


Figure 4.9: Distribution of the images in the 66 013 images part of the dataset by its place category.

number of images added from the Places365 Database depend on the category. The idea was to match the number of every place category and try to add the same amount of images with no incidents. For example, for the category *house* there are 1898 images according to Figure 4.9, so 1898 images from the Places365 Database of the class *house* were added, assuming that none of the images from Places365 contained an incident.

For the mixed category *(no place, no disaster)*, images from ImageNet [4] were added.

Another interesting figure to show of the dataset is a matrix where the rows correspond to the incident categories and the columns correspond to the places categories and try to see if we have a balanced dataset between all the mixed categories or which of them have more images or less. To have a better view, every point of the matrix follows this expression:

$$A[x, y] = log(1 + n) \tag{4.1}$$

where n represents the number of images in the mixed category *(x, y)*. The matrix is shown in Figure 4.10

There are some points worth to be highlighted. The point in the matrix with a higher value is the *(volcano, volcanic eruption)* mixed category. This is because, as explained in Section

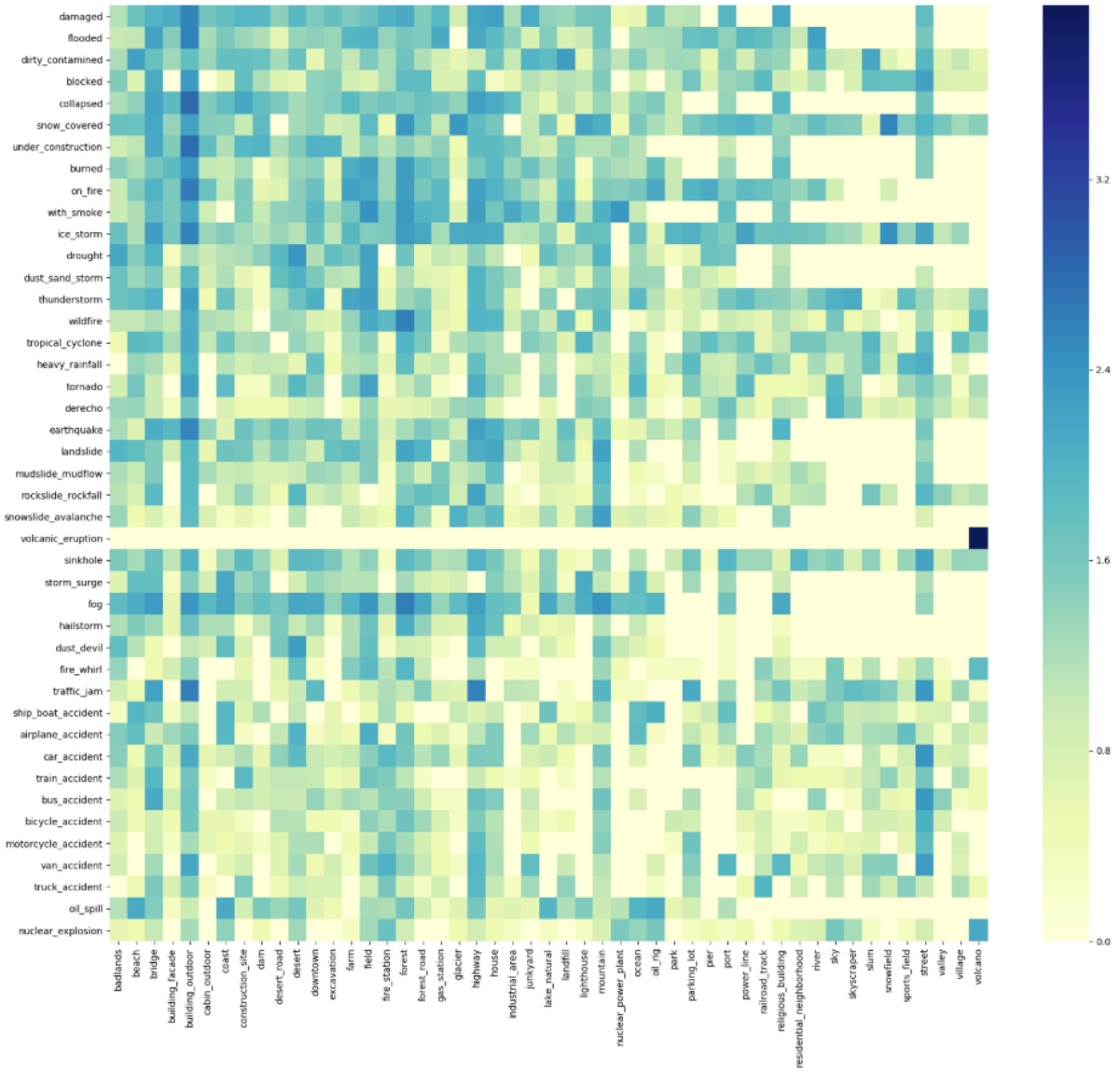


Figure 4.10: Distribution of the images in the dataset shown in a heat matrix.

4.2.3, we had almost 8000 images with the label *volcanic eruption* and all of them were labeled as *volcano*. On the other hand, all the other mixed categories only contain, at most, 400 images. A column important to highlight is the *building outdoor* column. This was the category for which we downloaded more images because we used a lot of synonyms, therefore it makes sense that we have a lot of images for this category in every incident. Additionally, the place categories such as *river*, *sky*, *skyscraper*, *slum* or *snowfield* are among the ones with less images as it is shown in the right columns of the matrix. For this classes we only have images with major incidents like *snow covered* or *thunderstorm*. Finally the second and third highest value points are the ones that contain the incident *traffic jam* in the place categories *highway* and *street* as they are one of the most common incidents nowadays.

5 Incidents and Places Classifier

After building the dataset, in this chapter, we are going to explore the image classification task. Firstly, the model used will be explained. Secondly, the training task will be discussed. And finally, the results of the classification will be presented.

5.1 Model

The architecture used for the classification task is ResNet [9], a model introduced by *Microsoft Research* on 2016. ResNet is a type of residual neural network that can have different number of layers. For this case, two structures have been used: an 18-layer ResNet and a 50-layer ResNet.

Independently of the number of layers, a ResNet is composed of different stages of convolutional layers and activation functions as well as a fully connected layer at the end. For instance, an 18-layer ResNet has a 7x7 filter convolutional layer, 16 3x3 filter convolutional layers and a fully connected layer at the end. It also uses $ReLU$ as the activation function. The 50-layers network has a $7x7$ filter convolutional layer, 32 1x1 filter convolutional layers, 16 3x3 filter convolutional layers and a fully connected layer. It also uses $ReLU$ as the activation function.

The main structure of the model is the following (Figure 5.1):

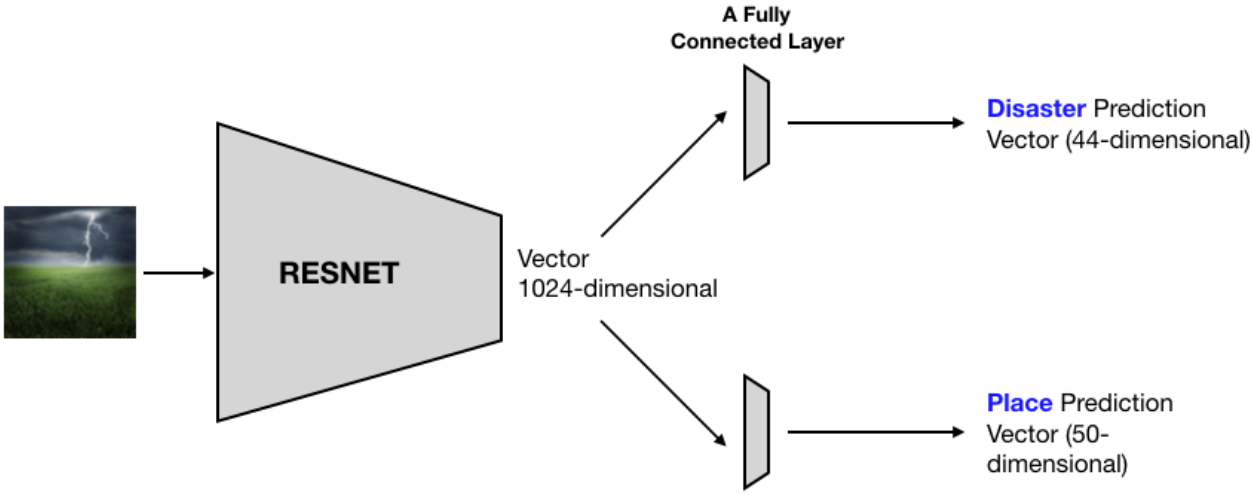


Figure 5.1: Model used to perform the experiments.

The input of the model is an image. After that, there is an 18 or 50 layers ResNet where the output information is saved into a 1024-dimensional vector. Then, the model is divided into two main parts. The incident path and the place path. The incident path contains a

fully connected layer that has a 1024-dimensional vector as the input and a 44-dimensional vector as the output. This 44 dimensions correspond to the 43 incident categories plus the new category *no disaster*. The place path is composed of a fully connected layer that has a 1024-dimensional vector as the input and a 50-dimensional vector as the output. These 50 dimensions correspond to the 49 incident categories plus the new category *no place*.

5.2 Training the classifier

For the training we had to divide the images in the database into three sets: the *training set* and the *validation set*, which are used in this training task, and the *test set* which we will use when testing the classifier. This splitting was computed trying to have the same amount of images for every *(incident, place)* category. For every mixed category a 70% of images was included in the *training set*, a 10% in the *validation set* and the last 20% in the *test set*.

For the training task, a 224x224 crop is randomly sampled from an image or a horizontal flip, with the per-pixel mean subtracted. Batch normalization [10] is adopted right after each convolution and before activation.

We use Stochastic Gradient Descent (SGD) as the optimizer with a mini-batch size of 250, a weight decay of 0.0001 and a momentum of 0.9. The learning rate starts from 0.01 and is divided by 10 every 10 epochs. We trained the classifier for 40 epochs. One epoch is completed when the whole dataset is passed forward and backward through the neural network.

As a loss function, we use the cross entropy loss as follows:

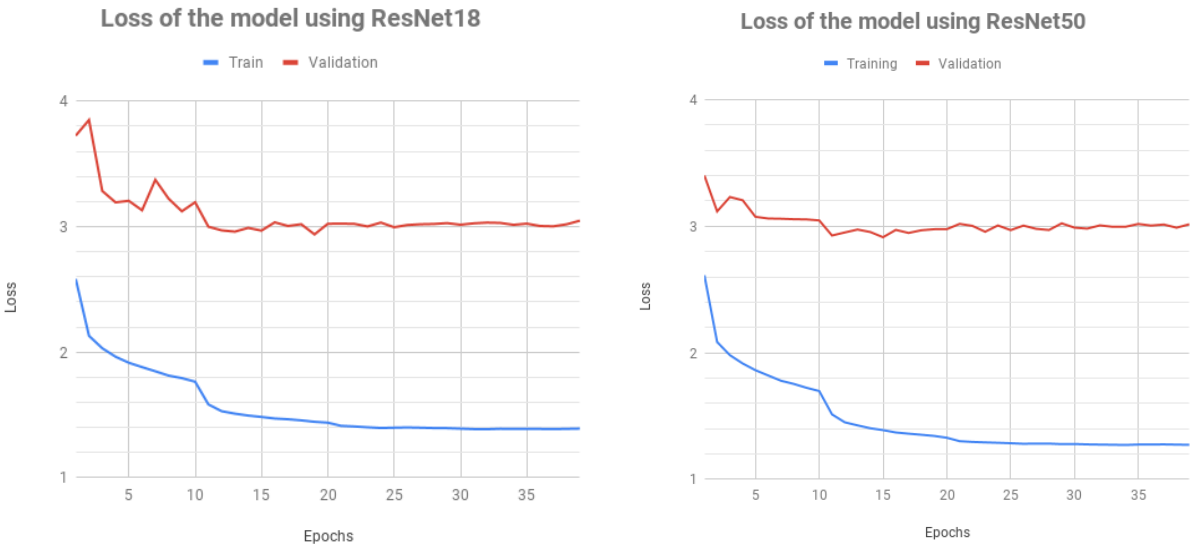
$$L = H_{\text{Incident}} + H_{\text{Place}} = -\sum_{c=1}^{\text{N}} y_{o,c} \ln\left(p_{o,c}\right) - \sum_{c=1}^{\text{M}} z_{o,c} \ln\left(q_{o,c}\right) \tag{5.1}$$

where N is the number of incidents classes, M the number of place classes, $y_{o,c}$ and $z_{0,c}$ are binary indicators (either 0 or 1) if the class label c is the correct classification for observation o and $p_{o,c}$ and $q_{o,c}$ are the predicted probabilities that observation o is of class c.

In addition, the architecture has not been trained with random weights at the beginning. A pre-trained model from Places365 [3] has been used to initialize the weights except for the last fully connected layer which we removed and replaces for one that fit the dimensions of our model. The weights of the fully connected layers in the incidents and the places paths have been randomly initialized.

First, we trained the model shown in Figure 5.1 using the 18-layer ResNet. The training and validation curves are illustrated in Figure 5.2a, where the convergence of the model can be

observed. The behavior is the expected as the curves start decreasing in a high pace and then flatten. In the training curve it is very visible how the learning rate is updated every 10 epochs since in the 10^{th} epoch there is a big change in the curve. Also, in the 20^{th} epoch there is the same change while in the 30^{th} epoch it is almost imperceptible. We can also see that the validation curve starts decreasing, but after some epochs it becomes quite stable and does not improve.



(a) ResNet18 used.

(b) ResNet50 used.

Figure 5.2: Plot of the loss in the training set and the validation set every epoch in the two-experiment set-up used.

Secondly, we trained the model shown in Figure 5.1 using the 50-layer ResNet. The training and validation curves are illustrated in Figure 5.2b. In this case the behavior is also as expected. The curves are very similar to the ones in the 18-layer case but the main difference is that the value of the training loss is lower with this architecture because it settles at 1.27 whereas in the first case it settles at 1.40. This feature was likely because the deeper the ResNet is, the better results it achieves in terms of loss and accuracy.

After training these two models, its quality was tested using the test set and the results are presented in the next section.

5.3 Results of the classification

The results of the classification are presented in Table 5.1. The metric used for evaluation is the accuracy:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{5.2}$$

The Top-1 accuracy is the percentage of the testing images where the top predicted label exactly match the ground-truth label. The Top-5 accuracy is the percentage of testing images where the ground-truth label is among the top ranked 5 predicted labels given by the neural network.

		Accuracy on the validation set	**Accuracy on the test set**	
		Top1	*Top1*	*Top5*
ResNet 18	*Incident*	72.50%	67.63%	92.75%
	Place	53.17%	49.17%	78.69%
ResNet 50	*Incident*	72.81%	69.87%	93.96%
	Place	54.92%	51.25%	81.10%

Table 5.1: Accuracy of two different models in the validation and test set.

First of all, the main attribute to comment is that the values for the incidents are much higher than the values for the places. That is because when training we have used approximately 100 000 images with an incident label but only a 45 000 images with a place label[4]. A network performs better with more data and we have 50% more data for the incidents.

Another aspect is that the results for the deeper ResNet are better, achieving an error rate of 31.13% in the Top-1 accuracy and 6.04% in the Top-5 accuracy for incidents. On the other hand, the 18-layers ResNet accomplishes error rates of 32.37% in the Top-1 accuracy and 7.25% in the Top-5 accuracy for incidents. This is a predictable behavior because the deeper the ResNet is, the better error rates it achieves.

In Figure 5.3 there are some examples of the results we got using the 50-layers classifier. The figure shows the images with its ground truth labels, the top 5 results for the incidents and the top 5 results for the places. All the results have its probability scores.

The first image we see in Figure 5.3 is a beach affected by an oil spill. The results in this picture are very good as the classifier detects the incident and the place with a high probability. The second image is a building affected by an earthquake but the classifier detects the incident *collapsed* with higher score than *earthquake* since earthquake is a quite wide concept difficult to extract from an image. The place predicted is *downtown* but with a

[4]This numbers correspond to a 70% of the total number of images as they are the ones in the *training set*


Ground Truth: oil spill
beach

TOP 5 disasters:
0.86331 -> oil_spill
0.04322 -> landslide
0.02649 -> storm_surge
0.02563 -> fog
0.01197 -> snow_covered

TOP 5 places:
0.87487 -> beach
0.11937 -> coast
0.00562 -> ocean
0.00005 -> desert
0.00003 -> snowfield


Ground Truth: earthquake
building_outdoor

TOP 5 disasters:
0.61908 -> collapsed
0.21841 -> earthquake
0.07951 -> train_accident
0.02958 -> under_construction
0.00990 -> burned

TOP 5 places:
0.45474 -> downtown
0.32537 -> building_outdoor
0.10819 -> construction_site
0.03829 -> building_facade
0.02285 -> industrial_area


Ground Truth: thunderstorm
field

TOP 5 disasters:
0.97729 -> thunderstorm
0.01718 -> tornado
0.00481 -> derecho
0.00044 -> dust_sand_storm
0.00015 -> tropical_cyclone

TOP 5 places:
0.70242 -> field
0.20913 -> sky
0.05439 -> farm
0.00707 -> highway
0.00608 -> power_line

Figure 5.3: Example of some of the results obtained with the 50-layers ResNet.

similar score as *building outdoor* which is a typical confusion that we will see in the confusion matrices. In the third picture the classifier predicts correctly a thunderstorm in a field.

One of the most interesting visualizations when carrying out a classification task is the confusion matrix, which is a summary of the prediction results of our model. The rows and the columns represent the categories. For our database, we will represent two confusion matrices: the incidents confusions and the places confusions. The values in the diagonal of the matrix correspond to the images classified as class x by the model and that have a ground truth value of x. We expect these values to be as high as the accuracy. The other values that stand out will represent confusions in the classifier. The matrix is normalized and represented as a heat map.

In Figure 5.4 the incidents confusion matrix is shown. Some of the main confusions are going to be commented bellow. The letters represent the main confusions and can be seen in the Figure 5.4 in red:

A- *(derecho, thunderstorm)*: a derecho is a type of cloud similar to the clouds that appear during thunderstorms.

B- *(van accident, car accident)*: usually in van accidents there are cars involved.

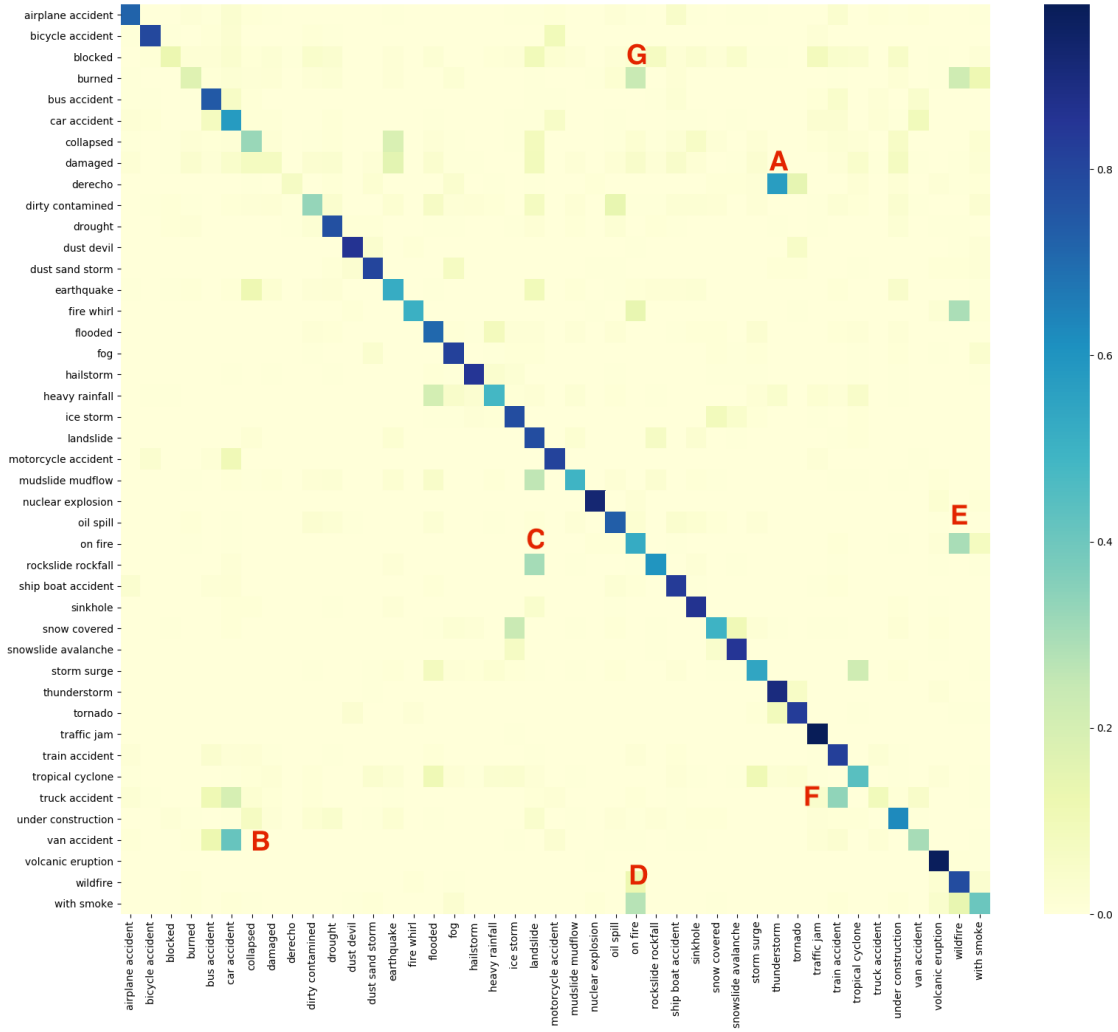


Figure 5.4: Incidents confusion matrix for the ResNet 50 model.

C- *(rockslide-rockfall, landslide)*: those are two incidents that overlap because in many landslides there are rocks and vice versa.

D- *(with smoke, on fire)*: an image can share both incidents.

E- *(on fire, wildfire)*: these two categories overlap and some images can share these two incidents.

F- *(truck accident, train accident)*: this confusion can be due to the fact that a truck and a train are similar vehicles.

G- *(burned, on fire)*: these two incidents overlap sometimes, but it is not a strong confusion.

On the other hand, in Figure 5.5 the places confusion matrix is shown. Some of the main confusions are going to be commented bellow. The letters represent the main confusions and can be seen in the Figure 5.4 in red:

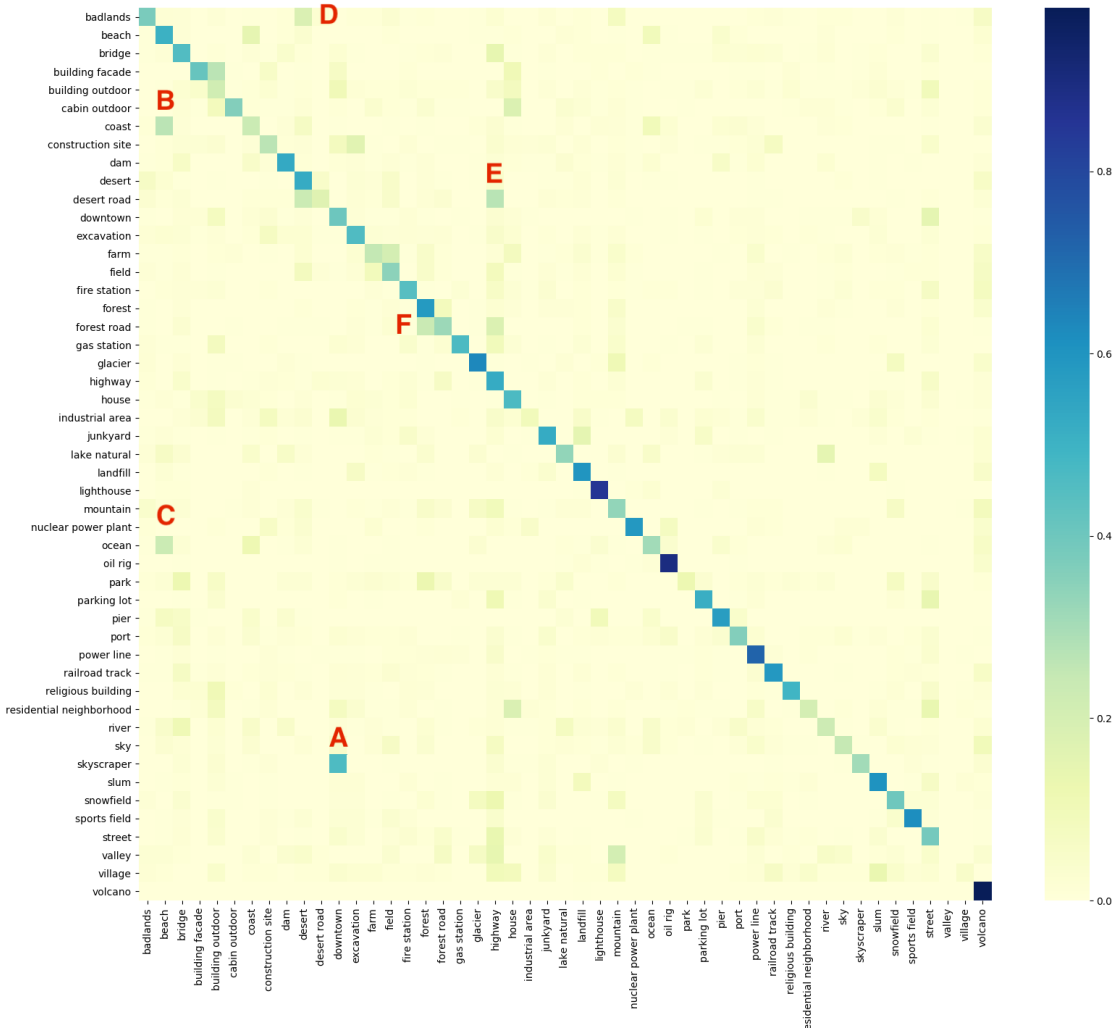


Figure 5.5: Places confusion matrix for the ResNet 50 model.

A- *(skyscraper, downtown)*: many images of skyscrapers can be labeled as downtown as those are similar scenarios.

B- *(coast, beach)*: these two categories overlap in many images and the differences between them are not very clear even for humans.

C- *(ocean, beach)*: this can be a confusion because the main element of the classes is the water.

D- *(badlands, desert)*: these two classes share many elements and this is a logical confusion.

E- *(desert road, highway)*: this is an interesting confusion that might be because of many highways are places in dry environments.

F- *(forest road, forest)*: this confusion appears due to the fact that these classes overlap as both have trees.

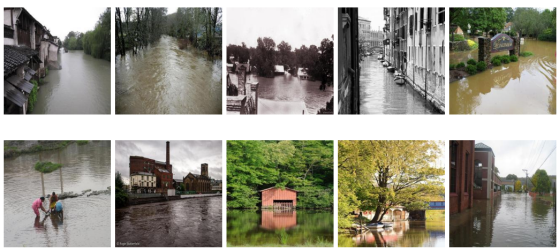
To sum up, all of the confusions explained are due to overlapping classes or similar elements in the images and we can conclude that the classifier does a good job and that the confusion are coherent and similar to the one that humans may have.

6 Applications

When having large datasets of labeled images, many applications can be developed, and two simple ones will be described. The first one being a consequence of asking ourselves: how many incidents can we find on scene-centric database such as Places365? The second one came from the database Flickr100 Database which has images with geo coordinates.

6.1 Are there any incidents in the Places365 Database?

For this experiment we applied the best classifier we had to the test set of Places365 Database [18]. The test set of the dataset has 73 000 images and 47 569 were classified with the top-1 label of *no disaster*. That means that it found that $34,18\%$ of the pictures had some type of incident. This value was slightly higher than expected and some visualizations were made in order to check which images have incidents (Figure 6.1).



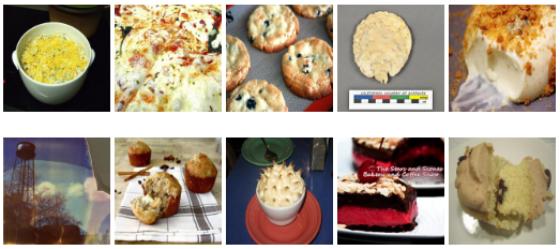
(a) Flooded.

(b) Burned.

(c) Fog.

(d) Airplane Accident.

(e) Dirty-Contamined.



(f) Nuclear Explosion.

Figure 6.1: Random sample of images labeled with some incidents in the Places365 database.

In Figure 6.1 six different disasters are shown. We can see a representation of images labeled as *flooded*, *burned*, *fog*, *airplane accident*, *dirty-contamined* and *nuclear explosion*. The first three classes in 6.1a, 6.1b and 6.1c have almost all samples correct. Some rivers get labeled as *flooded* and a white wall as *fog* but most of them look correct.

The *dirty-contamined* incident (Figure 6.1e) has half of the pictures correctly labeled. However, in 6.1d we can see that all images labeled as *airplane accident* contain planes in a good state so the network is classifying all images with airplanes as *airplane accidents*. The most surreal case is the one shown in 6.1f, were images or certain kinds of food like a pizza are labeled as a *nuclear explosion*. With experiments like this we can see the flaws in our database and we can work out on how to solve them.

6.2 Geo-localizing incidents

The Flickr100 Database has 50 million images and approximately 40% of them have geo-localization (latitude and longitude coordinates). Not all images are places, a lot of them are flowers, people, food or fireworks. From this huge dataset of images with coordinates, we downloaded a set of 50 000 (the ones that had geo-localization and a working URL) and forwarded them through our network to try to localize disasters in the globe. In Figure 6.2 there is a map showing a point for every image we have used.

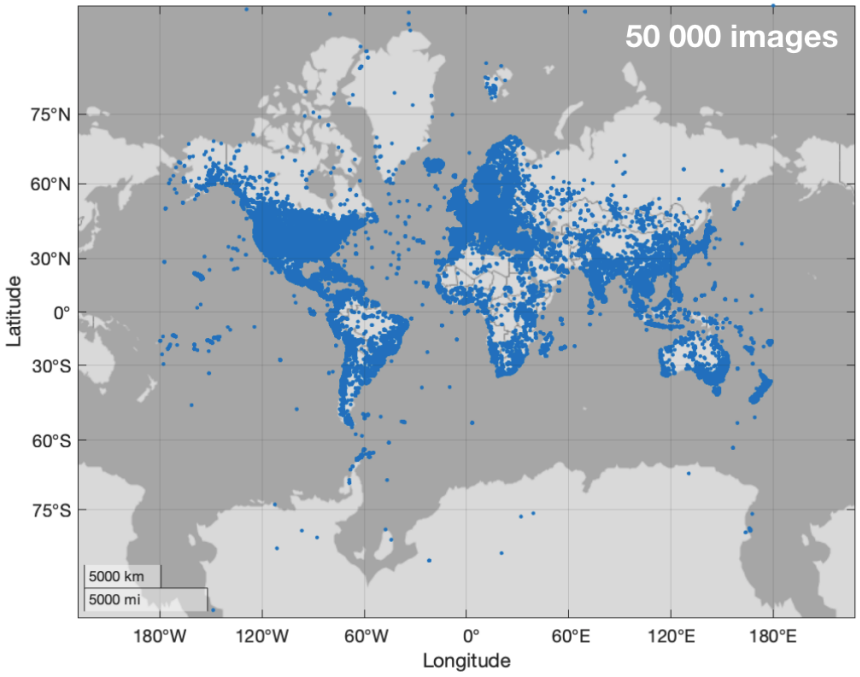


Figure 6.2: Map with the localization of 50 000 images from the Flickr100 Database.

With only 50 000 images, we have a small set to use for finding incidents. It found few

incidents but two of them are worth to mention. The first one is *traffic jam*, shown in Figure 6.3. The interesting thing about this map is that all the points in North America correspond to big cities such as New York or Los Angeles and the same happens in Europe. In the edges of the map there is a sample of images labeled with the class *traffic jam*.

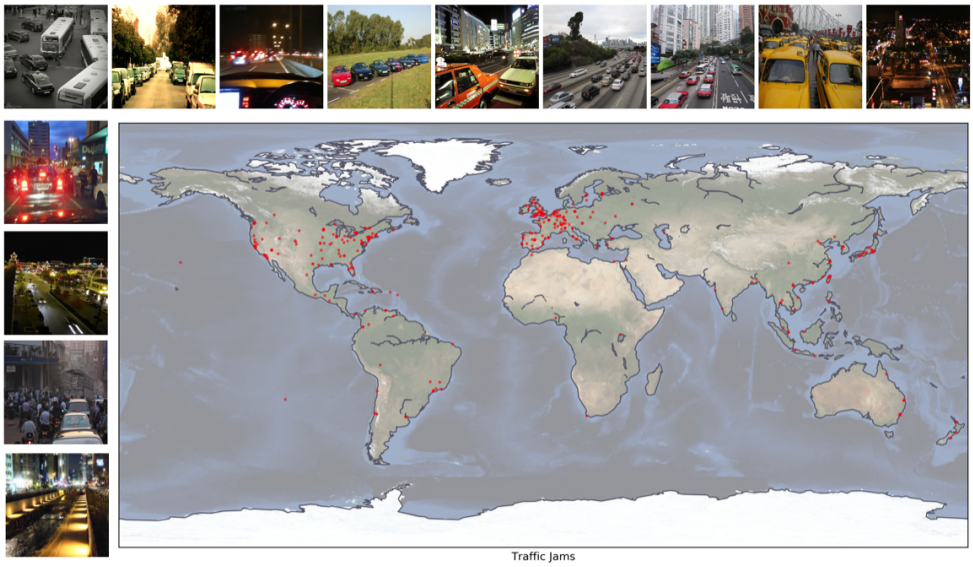


Figure 6.3: Map with the localization of traffic jams and some of the images labeled.

The second map (Figure 6.4) corresponds to images labeled as *volcanic eruption*. The first mistake we can see is that some fireworks are classified as having this incident. We have to bear in mind that this event is not quite common, so with only 50 000 images it is rather difficult to find this type of pictures. The points to highlight in this map are three volcanic archipelagos: Hawaii, the Canary Island and Iceland.

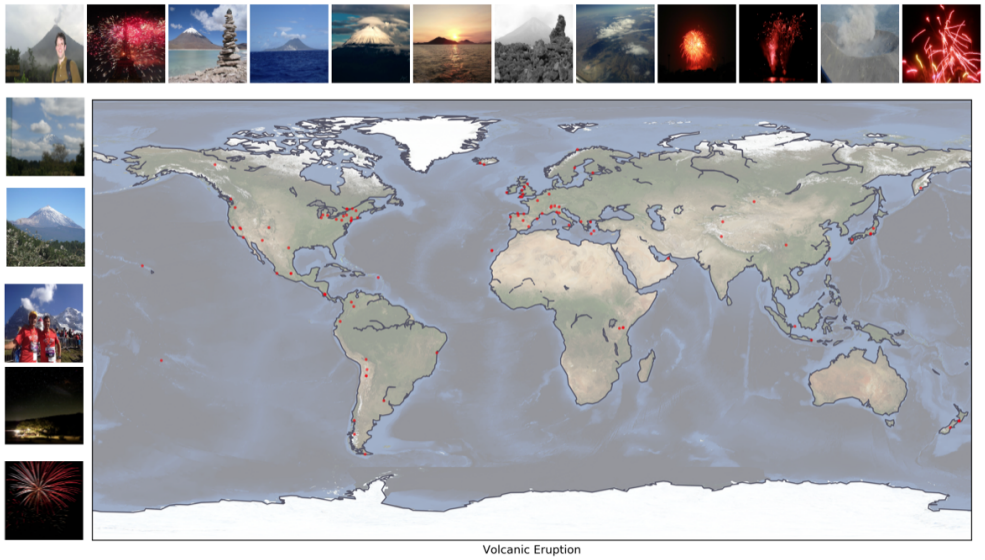


Figure 6.4: Map with the localization of volcanic eruption and some of the images labeled.

This shows us that with many more images and a better-trained classifier we can represent any event happening around the world. To achieve this we need to train the classifier with more *no disaster* examples to make sure that normal images are not labeled as false positives.

7 Conclusions

In this work we have presented the database *Incidents*, the Incidents and Places Classifier and some applications of the dataset. We have gathered and labeled almost 200 000 images of 43 different disasters happening in 49 different places. The dataset has been used to train a classifier which has achieved error rates of 30% in the top-1 prediction with 50-layers ResNet. Many conclusions can be drawn from this project.

First and foremost, deep learning models have many issues when trying to learn complex concepts. It is known that when using CNNs to identify objects or faces, the performances reach very high values, values that sometimes exceeded the human's accuracy. In contrast, when the network is trying to learn difficult concepts which are not associated completely to some visual trait like a scenario, an incident or even emotions on faces, the performances are not as good as one would expect. In our classification we have all kinds of concepts: some of them can be connected to some object, for example, a *fire whirl* that is a whirlwind induced by a fire often made up of flames. So when the network sees this object, it classifies the incident without doubt. Some other concepts such as *earthquake* or *blocked* are more complex incidents. For these classes, the pictures detected are destroyed buildings for *earthquake* or small streets with stairs and demonstrations for *blocked*. It is hard, even for a human, to reckon only from an image if an earthquake happened so we rely on the consequences of this natural disaster to classify this category. For blocked we believe that the network is classifying images that have small spaces to pass or images with many people on the center. To improve these classes we need a more diverse and complete database with many more examples.

Moreover, diversity in a database is one of the most important traits. Our high results on accuracy are due to the fact that all incidents we have are occurring in many different locations. But even though we have achieved this diversity in most of our categories, there are some that may need more examples. An interesting one is *thunderstorm*, where almost all pictures that we have are lightnings on a nightly background. That means that if our classifier sees a daytime thunderstorm it may not be able to recognize it correctly.

Furthermore, we have shown that a database with 200 000 images can perform considerably well in the image recognition task. If we take as an example the results obtained by Zhou et al. with the 10 million images dataset Places365 [18], we reach pretty similar results. We have to bear in mind that we can compare these results as the classification task is rather alike, but the number of training examples and categories in *Incidents* is a small fraction of Places365. The results resemble the ones in [18] where they achieve an accuracy of 55.29%. Our results are 3 to 4 points lower for the places and 15 points higher for the incidents, even

though our dataset has only 200 000 images compared to the 10 million in Places365.

Also, after seeing the results obtained in Section 6.1, we can understand the importance of having a reliable disaster/no-disaster detector. If applications shall be developed from this dataset it is important to add many *no disaster* images as we do not want a model detecting accidents every time it sees a plane or a train.

Finally, many applications can be developed from *Incidents*. Some of them have been shown in Section 6 but more complex applications can be developed as well. One of the most interesting ones is the use of Generative Adversarial Networks (GAN) [8] for removing or creating incidents in an image. With algorithms like CycleGAN [19] and examples of places with incidents and without incidents, a network that removes and adds disasters could be trained. This is a tough task for now, since we do not have enough images to achieve trustworthy results.

In conclusion, the *Incidents* database not only has been able to train a successful classifier, but also can be used for many applications related to humanitarian assistance. With this type of databases we are capable of showing that deep learning applied to computer vision can be helpful in using artificial intelligence for good.

References

[1] AI for Good: Humanitarian Action. `https://www.microsoft.com/en-us/ai/ai-for-humanitarian-action`. Accessed: 2019-03-29.

[2] CS231n: Convolutional neural networks for visual recognition. `http://cs231n.github.io/`. Accessed: 2019-05-10.

[3] Github respository of places365. `https://github.com/CSAILVision/places365`. Accessed: 2019-05-10.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[5] Pedro M Domingos. A few useful things to know about machine learning. *Commun. acm*, 55(10):78–87, 2012.

[6] Jigar Doshi, Saikat Basu, and Guan Pang. From satellite imagery to disaster insights. *arXiv preprint arXiv:1812.07033*, 2018.

[7] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[11] Mohammad Jahanian, Yuxuan Xing, Jiachen Chen, KK Ramakrishnan, Hulya Seferoglu, and Murat Yuksel. The evolving nature of disaster management in the internet and social media era. In *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pages 79–84. IEEE, 2018.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Núria Marzo. Incidents categories, example images and definitions. `http://wednesday.csail.mit.edu/wednesday/gt_html_scripts/gt_disasters.html`. Accessed: 2019-05-14.

[16] Núria Marzo. Places categories, example images and definitions. `http://wednesday.csail.mit.edu/wednesday/gt_html_scripts/gt_places.html`. Accessed: 2019-05-14.

[17] P Mass, C Nayak, A Dow, A Gros, W Mason, IO Filiz, C Duik, G Burrows, MC Jackman, V Sharma, C Lang, W Malik, and D Patel. Facebook disaster maps: Methodology. `https://research.fb.com/facebook-disaster-maps-methodology/`. Accessed: 2019-03-29.

[18] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Appendices

A Old incident categories

In this section the first set of incident categories divided by type is shown in the Table A.1.

Type	Old Categories
Damaged	damaged, flooded, destroyed, dirty, contaminated, blocked, low visibility, collapsed, frozen, under construction, dangerous and burned
Natural Disaster	cold wave, heat wave, winter storm, blizzard, thunder-snow, ice storm, drought, heat burst, dust storm, thunderstorm, firestorm, cyclone, typhoon, hurricane, tornado, lighting, haboob, derecho, strong wind, heavy rain, earthquake, landslide, mud-flow, volcanic eruption, snow avalanche, sinkhole, quick-stand, coastal flood, river flood, tsunami, cloudburst, high waves, wild fire, epidemic outbreak, pandemic outbreak, famine, fire, fog, flood, rock-slide, snow-slide, landslide, insect infestation, mudslide, locust infestation, snow storm, storm and storm surge
Man-Made Incident	attack, terrorist attack, shooting, fight, violence, riot, military action, armed conflict, war, police activity, police pursuit, military march, protest, kidnapping, arrest, assault, harassment, intimidation, forced recruitment, robbery, killing, demolition, construction, reparation, traffic jam, rescue operation, aid distribution and humanitarian help
Transport. & Nuclear Accidents	railway accident, railway disaster, railway crash, railway wreck, railway collision, railway explosion, railway incident, railway flood, maritime accident, maritime disaster, maritime crash, maritime wreck, maritime collision, maritime explosion, maritime incident, maritime flood, plane accident, plane disaster, plane crash, plane wreck, plane collision, plane explosion, plane incident, plane flood, car accident, car disaster, car crash, car wreck, car collision, car explosion, car incident, car flood, train accident, train disaster, train crash, train wreck, train collision, train explosion, train incident, train flood, tram accident, tram disaster, tram crash, tram wreck, tram collision, tram explosion, tram incident, tram flood, bus accident, bus disaster, bus crash, bus wreck, bus collision, bus explosion, bus incident, bus flood, bicycle accident, bicycle disaster, bicycle crash, bicycle wreck, bicycle collision, bicycle explosion, bicycle incident, bicycle flood, motorcycle accident, motorcycle disaster, motorcycle crash, motorcycle wreck, motorcycle collision, motorcycle explosion, motorcycle incident, motorcycle flood, van accident, van disaster, van crash, van wreck, van collision, van explosion, van incident, van flood, ship accident, ship disaster, ship crash, ship wreck, ship collision, ship explosion, ship incident, ship flood, boat accident, boat disaster, boat crash, boat wreck, boat collision, boat explosion, boat incident, boat flood, track accident, track disaster, track crash, track wreck, track collision, track explosion, track incident, track flood, public transportation accident, public transportation disaster, public transportation crash, public transportation wreck, public transportation collision, public transportation explosion, public transportation incident, public transportation flood, oil spill, levee breach, nuclear accident, nuclear explosion, nuclear incident and chemical spill
People Attributes	bleeding people, attacked people, assaulted people, beaten people, punched people, abused people, poor people, malnourished people, homeless people, dead people, sick people, shoot people, injured people, drown people, held people, imprisoned people, expelled people, mistreated people, hostile people, aggressive people, armed people, murdering people, punching people, shooting people, attacking people and robbing people

Table A.1: Old categories grouped by type.

B Dataset

B.1 Incidents

Some example of the categories are going to be shown. It will include the category name, a definition and some example images:

- **Traffic Jam**: a line (or several lines) of stationary or very slow-moving traffic, caused by roadworks, an accident, or heavy congestion.

Figure B.1: Traffic Jam examples.

- **Fire Whirl**: whirlwind induced by a fire and often made up of flame or ash.

Figure B.2: Fire Whirl examples.

- **Thunderstorm**: type of storm characterized by the presence of lightning and its acoustic effect on the Earth's atmosphere, known as thunder.

Figure B.3: Thunderstorm examples.

- **Ice Storm**: type of storm characterized by freezing rain, which results in accumulation of at least 0.25-inch of ice on exposed surfaces.

Figure B.4: Ice Storm examples.

- **Dirty-Contamined**: with garbage, unclean, or toxic.

Figure B.5: Dirty-Contamined examples.

Examples for the rest of the 43 categories can be found in [15].

B.2 Places

Some example of the categories are going to be shown. It will include the category name, a definition and some example images:

- **Sky**: the upper atmosphere as seen from the earth's surface.

Figure B.6: Sky examples.

- **Desert**: a dry region with little rainfall and extreme temperatures, which is covered in sand and can have vegetation.

Figure B.7: Desert examples.

- **Forest**: large area covered chiefly with trees and undergrowth than can contain paths.

Figure B.8: Forest examples.

- **Railroad Track**: a road on which trains run, composed of parallel steel rails supported by ties.

Figure B.9: Railroad track examples.

- **Skyscraper**: a very tall modern building, usually in a city.

Figure B.10: Skyscraper examples.

Examples for the rest of the 49 categories can be found in [16].

C Query vs Places365 Matrix

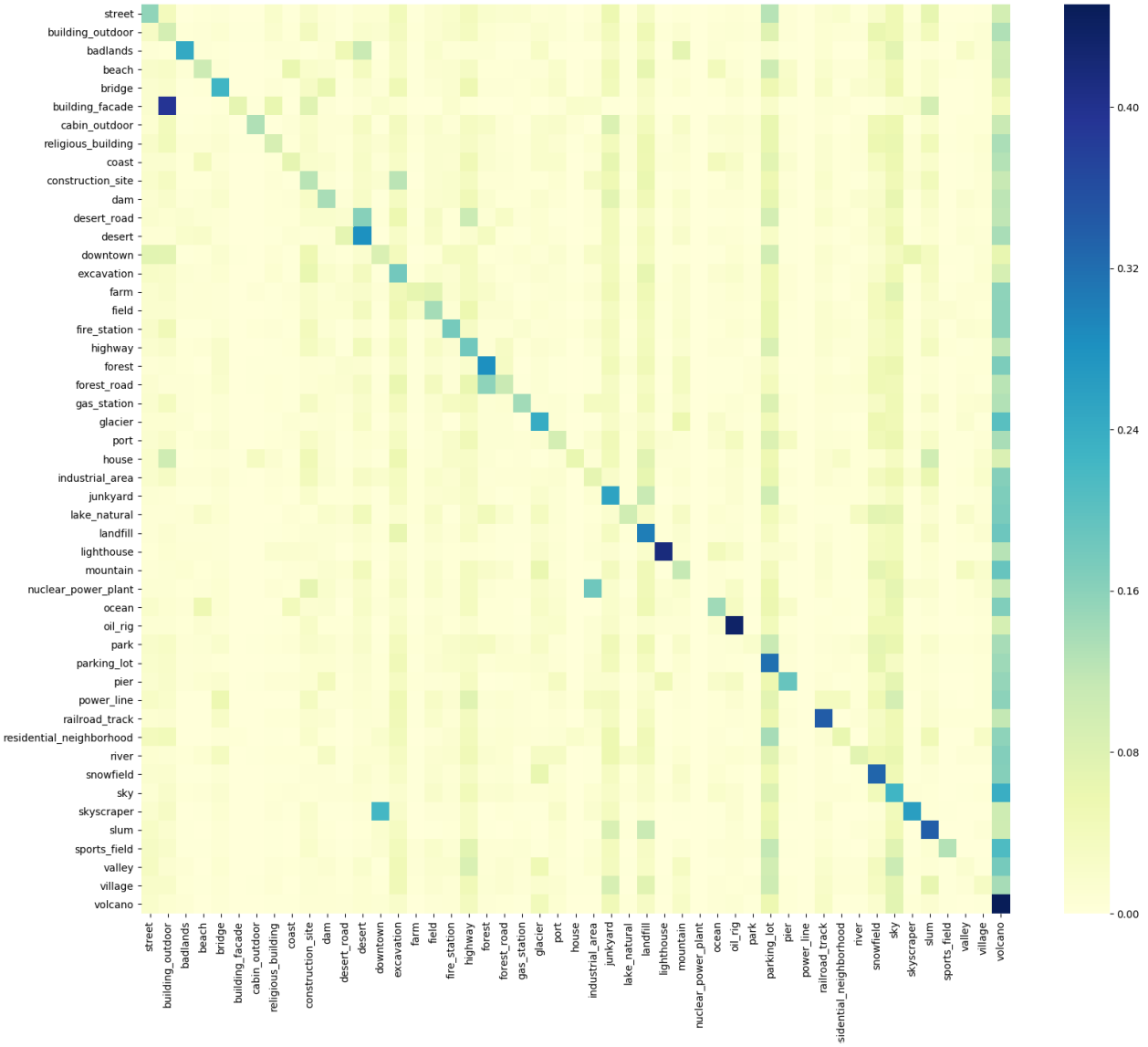


Figure C.1: Matrix that for every image represents the query it was downloaded with vs the place label that the Place365 Classifier gives it.